



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/731,548	12/05/2003	Peter Szpak	MWS-101	4558
26694	7590	08/21/2007		
VENABLE LLP P.O. BOX 34385 WASHINGTON, DC 20043-9998			EXAMINER WATT, CHRIS A	
			ART UNIT 2174	PAPER NUMBER
			MAIL DATE 08/21/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

88

Office Action Summary	Application No.	Applicant(s)	
	10/731,548	SZPAK ET AL.	
	Examiner	Art Unit	
	Chris Watt	2174	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 08 June 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-33 and 35-39 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-33 and 35-39 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 05 December 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This communication is responsive to the Amendment filed 6/8/2007.
2. Claims 1-33 and 35-39 are pending in this application. Claims 1, 14, 16, 30, 31, 32 and 33 are the independent claims. In the instant amendment, claims 1, 7, 9, 14, 16, 17, 22, 24, 26, 27, 30, 33 35, 36, 37, 38 and 39 were amended and claim 34 was cancelled. No claims were added. This action is made non-final.
3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Claim Rejections - 35 USC § 103

4. Claims 1-33 and 35-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coad et al. ("Coad" US Patent No. 6,851,105) in view of Frid-Nielsen ("Frid-Nielsen" US Patent No. 5,399,433).

Regarding independent claim 1, Coad teaches a method for generating embedded code from a graphical model (i.e. compare graphical model and generated embedded code in FIGS. 9-12 et seq. of Coad), comprising the steps of: prompting a user to specify the embedded code (i.e. "Display pattern options corresponding to selected element type" in FIG. 2 et seq. of Coad). Coad does not teach generating code for a code generation goal.

Frid-Nielsen teaches at least one code generation goal (i.e. design of objects in FIG. 2A et seq. of Frid-Nielsen; see also FIG. 5) and generating code in a compliable form for the specified code generation goal (i.e. compiled and linked files in FIGS. 3A and 3B et seq. of Frid-Nielsen). It would have been obvious to an artisan at the time of

Art Unit: 2174

the invention to combine the code generation based on goals of Frid-Nielsen into the embedded code generation of Coad. Said artisan would have been motivated to combine Frid-Nielsen into Coad to create a structured and pre-arranged set of modes in which the user will accomplish the purpose of the program as established in the goals, while creating the added benefit of flexibility in accomplishing the goals for each specific user (i.e. see col. 6 line 41 et seq. of Frid-Nielsen).

Regarding dependent claim 2, Coad in combination with Frid-Nielsen teaches the method of claim 1, further comprising the step of modifying one or more parameters of the graphical model to comply with the code generation goal in response to the user specifying said at least one code generation goal (i.e. "configurable properties and parameters" in FIG. 2 et seq. of Coad).

Regarding dependent claim 3, Coad in combination with Frid-Nielsen teaches the method of claim 1, further comprising the step of providing feedback to the user regarding the compliance of the graphical model with a selected condition (i.e. "Are variations suitable" in FIG. 2 et seq. of Coad).

Regarding dependent claim 4, Coad in combination with Frid-Nielsen teaches the method of claim 3, wherein the user selects the selected condition through a user interface (i.e. see user selection of conditions in FIGS. 4-8 et seq. of Coad).

Regarding dependent claim 5, Coad in combination with Frid-Nielsen teaches the method of claim 4, wherein the user interface displays a list of conditions to be checked, and prompts the user to select one or more of the conditions (i.e. see user selection of conditions in FIGS. 4-8 et seq. of Coad).

Regarding dependent claim 6, Coad in combination with Frid-Nielsen teaches the method of claim 3, wherein the step of providing feedback to the user regarding the compliance of the graphical model with a selected condition comprises displaying a hyperlink for linking the selected condition to an object of the graphical model that does not comply with the selected condition (i.e. compare "Check variations for ... errors" in FIG. 15b et seq. of Coad with links in FIG. 5 et seq. of Frid-Nielsen).

Regarding dependent claim 7, Coad in combination with Frid-Nielsen teaches the method of claim 3, further comprising the step of modifying an object of the graphical model that does not comply with the selected condition (i.e. compare steps 212-214 in FIG. 2 et seq. of Coad, defining special patterns for modifying objects).

Regarding dependent claim 8, Coad in combination with Frid-Nielsen teaches the method of claim 7, wherein the step of modifying comprises identifying the object and prompting the user to manually modify a parameter of the object (i.e. modification of TObject parameters in FIG. 6B-7A et seq. of Frid-Nielsen).

Regarding dependent claim 9, Coad in combination with Frid-Nielsen teaches the method of claim 7, wherein the step of modifying comprises automatically modifying a parameter of the graphical model to comply with the selected condition (i.e. "Automatically ... to elements in pattern to pattern" in FIG. 15b et seq. of Coad).

Regarding dependent claim 10, Coad in combination with Frid-Nielsen teaches the method of claim 1, wherein the graphical model is a block diagram (i.e. FIG. 7G-7H et seq. of Frid-Nielsen; see also FIG. 9-12 et seq. of Coad).

Regarding dependent claim 11, Coad in combination with Frid-Nielsen teaches the method of claim 1, wherein each code generation goal corresponds to a general code generation goal (i.e. "pattern options corresponding to selected element type" in FIG. 2 et seq. of Coad).

Regarding dependent claim 12, Coad in combination with Frid-Nielsen teaches the method of claim 11, further comprising the step of prompting the user to specify at least one detailed code generation goal for each specified general code generation goal (i.e. does special property need to be defined" in FIG. 2 et seq. of Coad).

Regarding dependent claim 13, Coad in combination with Frid-Nielsen teaches the method of claim 12, further comprising the step of configuring the graphical model to comply with each detailed code generation goal (i.e. "pattern's configurable properties and parameters" in FIG. 2 et seq. of Coad).

Regarding independent claim 14, Coad teaches a method of preparing a graphical model for embedded code generation (i.e. compare graphical model and generated embedded code in FIGS. 9-12 et seq. of Coad), comprising the steps of: displaying a user interface for prompting a user and automatically changing parameters of the graphical model that are inconsistent (i.e. "Display pattern options corresponding to selected element type" in FIG. 2 et seq. of Coad). Coad does not teach generating code for a code generation goal.

Frid-Nielsen teaches a user specifying one or more code generation goals specified by the user (i.e. design of objects in FIG. 2A et seq. of Frid-Nielsen; see also FIG. 5). It would have been obvious to an artisan at the time of the invention to combine

Art Unit: 2174

the code generation based on goals of Frid-Nielsen into the embedded code generation of Coad. Said artisan would have been motivated to combine Frid-Nielsen into Coad to create a structured and pre-arranged set of modes in which the user will accomplish the purpose of the program as established in the goals, while creating the added benefit of flexibility in accomplishing the goals for each specific user (i.e. see col. 6 line 41 et seq. of Frid-Nielsen).

Regarding dependent claim 15, Coad in combination with Frid-Nielsen teaches the method of claim 14, further comprising the step of identifying a condition that does not comply with the code generation goals specified by the user (i.e. "Check variations for ... errors" in FIG. 15b et seq. of Coad). Coad does not teach generating code for a code generation goal.

Regarding independent claim 16, Coad teaches a method of preparing a graphical model for embedded code generation (i.e. compare graphical model and generated embedded code in FIGS. 9-12 et seq. of Coad), the method comprising the steps of: displaying a graphical user interface through which a user can specify the embedded code to be generated from the graphical model (i.e. "Display pattern options corresponding to selected element type" in FIG. 2 et seq. of Coad); providing feedback to the user regarding compliance of the graphical model (i.e. "Are variations suitable" in FIG. 2 et seq. of Coad). Coad does not teach at least one code generation goal for and response to a user specifying a code generation goal.

Frid-Nielsen teaches at least one code generation goal for and response to a user specifying a code generation goal (i.e. design of objects in FIG. 2A et seq. of Frid-

Nielsen; see also FIG. 5). It would have been obvious to an artisan at the time of the invention to combine the code generation based on goals of Frid-Nielsen into the embedded code generation of Coad. Said artisan would have been motivated to combine Frid-Nielsen into Coad to create a structured and pre-arranged set of modes in which the user will accomplish the purpose of the program as established in the goals, while creating the added benefit of flexibility in accomplishing the goals for each specific user (i.e. see col. 6 line 41 et seq. of Frid-Nielsen).

Regarding dependent claim 17, Coad in combination with Frid-Nielsen teaches the method of claim 16, further comprising the step of modifying one or more parameters of the graphical model to comply with the code generation goal in response to the user specifying said at least one code generation goal (i.e. "configurable properties and parameters" in FIG. 2 et seq. of Coad).

Regarding dependent claim 18, Coad in combination with Frid-Nielsen teaches the method of claim 16, wherein the step of providing feedback comprises indicating to the user whether the graphical model complies with a selected condition (i.e. "Are variations suitable" in FIG. 2 et seq. of Coad).

Regarding dependent claim 19, Coad in combination with Frid-Nielsen teaches the method of claim 18, wherein the user selects the selected condition through the user interface (i.e. see user selection of conditions in FIGS. 4-8 et seq. of Coad).

Regarding dependent claim 20, Coad in combination with Frid-Nielsen teaches the method of claim 19, wherein the user interface displays a list of conditions to be

Art Unit: 2174

checked, and prompts the user to select one or more of the conditions (i.e. see user selection of conditions in FIGS. 4-8 et seq. of Coad).

Regarding dependent claim 21, Coad in combination with Frid-Nielsen teaches the method of claim 18, wherein the step of indicating to the user whether the graphical model complies with a selected condition comprises displaying a hyperlink for linking the selected condition to an object of the graphical model that does not comply with the selected condition (i.e. compare "Check variations for ... errors" in FIG. 15b et seq. of Coad with links in FIG. 5 et seq. of Frid-Nielsen).

Regarding dependent claim 22, Coad in combination with Frid-Nielsen teaches the method of claim 18, further comprising the step of modifying an object of the graphical model that does not comply with the selected condition (i.e. compare steps 212-214 in FIG. 2 et seq. of Coad, defining special patterns for modifying objects).

Regarding dependent claim 23, Coad in combination with Frid-Nielsen teaches the method of claim 22, wherein the step of modifying comprises identifying the object and prompting the user to manually modify a parameter of the object (i.e. modification of TObject parameters in FIG. 6B-7A et seq. of Frid-Nielsen).

Regarding dependent claim 24, Coad in combination with Frid-Nielsen teaches the method of claim 22, wherein the step of modifying comprises automatically modifying a parameter of the graphical model to comply with the selected condition (i.e. "Automatically ... to elements in pattern to pattern" in FIG. 15b et seq. of Coad).

Regarding dependent claim 25, Coad in combination with Frid-Nielsen teaches the method of claim 16, wherein the graphical model is a block diagram (i.e. FIG. 7G-7H et seq. of Frid-Nielsen; see also FIG. 9-12 et seq. of Coad).

Regarding dependent claim 26, Coad in combination with Frid-Nielsen teaches the method of claim 16, further comprising the step of generating code that is compatible with said at least one code generation goal (i.e. "pattern options corresponding to selected element type" in FIG. 2 et seq. of Coad).

Regarding dependent claim 27, Coad in combination with Frid-Nielsen teaches the method of claim 16, wherein each code generation goal corresponds to a general code generation goal (i.e. "pattern options corresponding to selected element type" in FIG. 2 et seq. of Coad).

Regarding dependent claim 28, Coad in combination with Frid-Nielsen teaches the method of claim 27, further comprising the step of prompting the user to specify at least one detailed code generation goal for each specified general code generation goal (i.e. does special property need to be defined" in FIG. 2 et seq. of Coad).

Regarding dependent claim 29, Coad in combination with Frid-Nielsen teaches the method of claim 28, further comprising the step of configuring the graphical model to comply with each detailed code generation goal (i.e. "pattern's configurable properties and parameters" in FIG. 2 et seq. of Coad).

Regarding independent claim 30, Coad teaches, in a graphical modeling environment, a medium holding computer-executable instructions for a method, comprising the steps of: displaying a graphical user interface though which a user can

Art Unit: 2174

specify code to be generated from the graphical model (i.e. compare graphical model and generated embedded code in FIGS. 9-12 et seq. of Coad); and in response, providing feedback to the user regarding compliance of the graphical model (i.e. "Are variations suitable" in FIG. 2 et seq. of Coad). Coad does not teach at least one code generation goal target characteristic for a user specifying a code generation goal.

Frid-Nielsen teaches at least one code generation goal target characteristic for a user specifying a code generation goal (i.e. design of objects in FIG. 2A et seq. of Frid-Nielsen; see also FIG. 5). It would have been obvious to an artisan at the time of the invention to combine the code generation based on goals of Frid-Nielsen into the embedded code generation of Coad. Said artisan would have been motivated to combine Frid-Nielsen into Coad to create a structured and pre-arranged set of modes in which the user will accomplish the purpose of the program as established in the goals, while creating the added benefit of flexibility in accomplishing the goals for each specific user (i.e. see col. 6 line 41 et seq. of Frid-Nielsen).

Regarding independent claim 31, Coad teaches, in a graphical modeling environment, a medium holding computer- executable instructions for a method, comprising the steps of: displaying a user interface for prompting a user (i.e. "Display pattern options corresponding to selected element type" in FIG. 2 et seq. of Coad); and automatically changing parameters of the graphical model that are inconsistent with the code generation goals specified by the user (i.e. "Automatically ... to elements in pattern to pattern" in FIG. 15b et seq. of Coad). Coad does not teach a user specifying one or more code generation goals.

Frid-Nielsen teaches a user specifying one or more code generation goals (i.e. design of objects in FIG. 2A et seq. of Frid-Nielsen; see also FIG. 5). It would have been obvious to an artisan at the time of the invention to combine the code generation based on goals of Frid-Nielsen into the embedded code generation of Coad. Said artisan would have been motivated to combine Frid-Nielsen into Coad to create a structured and pre-arranged set of modes in which the user will accomplish the purpose of the program as established in the goals, while creating the added benefit of flexibility in accomplishing the goals for each specific user (i.e. see col. 6 line 41 et seq. of Frid-Nielsen).

Regarding independent claim 32, Coad teaches, in a graphical modeling environment, a medium holding computer- executable instructions for a method, comprising the steps of: prompting a user to specify for the embedded code (i.e. "Display pattern options corresponding to selected element type" in FIG. 2 et seq. of Coad). Coad does not teach at least one code generation goal and generating code in a compliable form for the specified code generation goal.

Frid-Nielsen teaches at least one code generation goal (i.e. design of objects in FIG. 2A et seq. of Frid-Nielsen; see also FIG. 5) and generating code in a compliable form for the specified code generation goal (i.e. compiled and linked files in FIGS. 3A and 3B et seq. of Frid-Nielsen). It would have been obvious to an artisan at the time of the invention to combine the code generation based on goals of Frid-Nielsen into the embedded code generation of Coad. Said artisan would have been motivated to combine Frid-Nielsen into Coad to create a structured and pre-arranged set of modes in

which the user will accomplish the purpose of the program as established in the goals, while creating the added benefit of flexibility in accomplishing the goals for each specific user (i.e. see col. 6 line 41 et seq. of Frid-Nielsen).

Regarding independent claim 33, Coad teaches an apparatus comprising: at least one processor; a memory coupled to the at least one processor; and a computer program residing in the memory and being executed by the at least one processor, wherein the computer program includes a wizard for guiding a user through a process for preparing a graphical model for a code generation process for creating code based on the graphical model (i.e. "Receive a selected element type" and "Display pattern options" in FIG. 2 et seq. of Coad). Coad does not teach at least one code generation goal specified by the user.

Frid-Nielsen teaches at least one code generation goal specified by the user (i.e. design of objects in FIG. 2A et seq. of Frid-Nielsen; see also FIG. 5). It would have been obvious to an artisan at the time of the invention to combine the code generation based on goals of Frid-Nielsen into the embedded code generation of Coad. Said artisan would have been motivated to combine Frid-Nielsen into Coad to create a structured and pre-arranged set of modes in which the user will accomplish the purpose of the program as established in the goals, while creating the added benefit of flexibility in accomplishing the goals for each specific user (i.e. see col. 6 line 41 et seq. of Frid-Nielsen).

Regarding dependent claim 35, Coad in combination with Frid-Nielsen teaches the apparatus of claim 33, wherein the wizard configures the graphical model based on

the at least one code generation goal specified by the user (i.e. design of objects in FIG. 2A et seq. of Frid-Nielsen; see also FIG. 5; see also "Receive a selected element type" and "Display pattern options" in FIG. 2 et seq. of Coad).

Regarding dependent claim 36, Coad in combination with Frid-Nielsen teaches the apparatus of claim 33, wherein the computer program generates code in compliance with the at least one code generation goal specified by the user (i.e. "Are variations suitable" in FIG. 2 et seq. of Coad).

Regarding dependent claim 37, Coad in combination with Frid-Nielsen teaches the apparatus of claim 33, wherein the wizard prompts the user to select one or more conditions to be checked in the graphical model (i.e. see user selection of conditions in FIGS. 4-8 et seq. of Coad).

Regarding dependent claim 38, Coad in combination with Frid-Nielsen teaches the apparatus of claim 37, wherein the wizard identifies objects in the graphical model that do not comply with the selected conditions (i.e. compare steps 212-214 in FIG. 2 et seq. of Coad, defining special patterns for modifying objects).

Regarding dependent claim 39, Coad in combination with Frid-Nielsen teaches the apparatus of claim 37, wherein the wizard modifies objects in the graphical model that do not comply with the selected conditions (i.e. modification of TObject parameters in FIG. 6B-7A et seq. of Frid-Nielsen).

Response to Arguments

Applicant's arguments with respect to claims 1-33 and 35-39 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chris Watt whose telephone number is (571) 270-1046. The examiner can normally be reached on Monday-Thursday 6:30-4:00 Eastern.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine L. Kincaid can be reached on (571) 272-4063. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Chris A. Watt/

CAW

August 14, 2007


KRISTINE KINCAID
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100